

Appendix

A. Implementation Details

A.1. Complete Algorithm

Training the StyleT2I framework contains two steps—Step 1: train the *Text-to-Direction* module (Algorithm 1); Step 2: train the *Attribute-to-Direction* module (Algorithm 2). The pseudocode of the inference algorithm of StyleT2I for synthesizing images conditioned on the given text is shown in Algorithm 3.

Algorithm 1: Train *Text-to-Direction* module

Input: G : pretrained generator, M_t : training iterations, $\mathcal{T} = \{\mathbf{t}\}$: training set of text.
Output: $\mathcal{F}_{\text{text}}$: *Text-to-Direction* module

```

1 for  $k : 1 \dots M_t$  do
2    $\mathbf{z} \sim \mathcal{W}+$  // random latent code
    sampled from  $\mathcal{W}+$  space
3    $\mathbf{t} \sim \mathcal{T}$  // text sampled from the
    training set
4    $\mathbf{s} = \mathcal{F}_{\text{text}}(\mathbf{z}, \mathbf{t})$  // predict sentence
    direction
5    $\mathbf{z}_s = \mathbf{z} + \mathbf{s}$  // text-conditioned code
6    $\hat{\mathbf{I}} = G(\mathbf{z}_s)$  // synthesize image
7    $\mathcal{L}_s = \mathcal{L}_{\text{contras}}(\hat{\mathbf{I}}, \mathbf{t}) + \mathcal{L}_{\text{norm}}(\mathbf{s})$  // compute
    loss
8    $\mathcal{F}_{\text{text}} \leftarrow \text{Adam}(\nabla_{\mathcal{F}_{\text{text}}} \mathcal{L}_s)$  // update  $\mathcal{F}_{\text{text}}$ 
9 return  $\mathcal{F}_{\text{text}}$ 

```

A.2. Hyperparameters and Network Architecture

We pretrain StyleGAN2 on each dataset (CelebA-HQ [18] and CUB [61]) with 300,000 iterations. In CLIP [47], we use ViT-B/32 [10] architecture as the image encoder. We use Adam optimizer [22] with 10^{-4} learning rate to train both modules. The *Text-to-Direction* module is trained with 60,000 iterations and the batch size is 40. The *Attribute-to-Direction* module is trained with 1000 iterations with batch size of 2. The architectures of *Text-to-Direction* module and *Attribute-to-Direction* module are shown in Fig. 10.

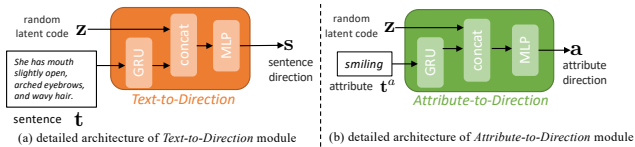


Figure 10. Detailed architectures of (a) *Text-to-Direction* module and (b) *Attribute-to-Direction* module.

Algorithm 2: Train *Attribute-to-Direction* module

Input: $\mathcal{V} = \{\mathbf{t}^a\}$: attribute vocabulary, G : pretrained generator, \mathcal{S} : weakly-supervised segmentation network, M_a : training iterations
Output: $\mathcal{F}_{\text{attr}}$: *Attribute-to-Direction* module

```

1 for  $m : 1 \dots M_a$  do
2    $\mathbf{z} \sim \mathcal{W}+$  // random latent code
    sampled from  $\mathcal{W}+$  space
3    $\mathbf{t}^a \sim \mathcal{V}$  // attribute sampled from
    vocabulary
4    $\mathbf{a} = \mathcal{F}_{\text{attr}}(\mathbf{z}, \mathbf{t}^a)$  // predict attribute
    direction
5    $\mathbf{z}_{\text{pos}} = \mathbf{z} + \mathbf{a}$  // positive latent code
6    $\mathbf{z}_{\text{neg}} = \mathbf{z} - \mathbf{a}$  // negative latent code
7    $\mathbf{I}_{\text{pos}}^a = G(\mathbf{z}_{\text{pos}})$  // positive image
8    $\mathbf{I}_{\text{neg}}^a = G(\mathbf{z}_{\text{neg}})$  // negative image
9    $\mathbf{M}^a = \mathcal{S}(\mathbf{I}_{\text{pos}}^a)$ 
    // pseudo-ground-truth mask
10   $\mathbf{I}_{\text{diff}}^a = \sum_c |\mathbf{I}_{\text{pos}}^a - \mathbf{I}_{\text{neg}}^a|$  // pixel-level
    difference
11   $\tilde{\mathbf{I}}_{\text{diff}}^a = \frac{\mathbf{I}_{\text{diff}}^a - \min(\mathbf{I}_{\text{diff}}^a)}{\max(\mathbf{I}_{\text{diff}}^a) - \min(\mathbf{I}_{\text{diff}}^a)}$  // min-max
    normalization
12   $\mathcal{L}_a = \mathcal{L}_{\text{semantic}}(\mathbf{I}_{\text{pos}}^a, \mathbf{I}_{\text{neg}}^a, \mathbf{t}^a) +$ 
     $\mathcal{L}_{\text{spatial}}(\tilde{\mathbf{I}}_{\text{diff}}^a, \mathbf{M}^a) + \mathcal{L}_{\text{norm}}(\mathbf{a})$  // compute
    loss
13   $\mathcal{F}_{\text{attr}} \leftarrow \text{Adam}(\nabla_{\mathcal{F}_{\text{attr}}} \mathcal{L}_a)$  // update  $\mathcal{F}_{\text{attr}}$ 
14 return  $\mathcal{F}_{\text{attr}}$ 

```

A.3. Attribute Extraction

On CelebA-HQ dataset, we use string matching to extract attributes from the text. For example, the word “bangs” in the sentence indicates the “bangs” attribute. On CUB dataset, we extract attributes based on part-of-speech (POS) tags and dependency parsing implemented in spaCy [16]. Concretely, given a text, we extract adjectives and nouns based on POS tags. Then, we leverage their dependency relations to extract the attributes. For example, in the text “the bird has a yellow breast,” “yellow” and “breast” has the adjectival modifier (amod) dependency relation, which indicates the “yellow breast” attribute. We also use other dependency relations to deal with sentences with more complex sentence structures. For example, in the text “the bird has a brown and yellow breast,” “yellow” and “brown” have the “conjunct” (conj) dependency relation, which indicates two attributes—“yellow breast” and “brown breast.”

Algorithm 3: Inference algorithm of StyleT2I

Input: G : pretrained generator, \mathbf{t} : input text,
 $\{\mathbf{t}_i^a\}_{i=1}^K$: extracted K attributes from text,
 $\mathcal{F}_{\text{text}}$: *Text-to-Direction* module, $\mathcal{F}_{\text{attr}}$:
Attribute-to-Direction module

Output: $\hat{\mathbf{I}}$: synthesized image conditioned on the
input text

```
1  $\mathbf{z} \sim \mathcal{W}+$  // random latent code
   sampled from  $\mathcal{W}+$  space
2  $\mathbf{s} = \mathcal{F}_{\text{text}}(\mathbf{z}, \mathbf{t})$  // predict sentence
   direction
3  $\mathbf{A} = \{\mathbf{a}_i \mid \cos(\mathbf{a}_i, \mathbf{s}) \leq 0\}$ . // set of
   attributes need to be adjusted
4  $\mathbf{s}' = \mathbf{s} + \sum_{\mathbf{a}_i \in \mathbf{A}} \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2}$  // adjust sentence
   direction
5  $\mathbf{z}_s = \mathbf{z} + \mathbf{s}'$  // text-conditioned code
6  $\hat{\mathbf{I}} = G(\mathbf{z}_s)$  // synthesize image
7 return  $\hat{\mathbf{I}}$ 
```

A.4. Pseudo-ground-truth Mask

We use [17] as a weakly-supervised part segmentation network to obtain pseudo-ground-truth masks. The network is a classifier supervised by binary attribute labels extracted from text. In specific, since each image is paired with multiple texts, we use the union of attributes extracted from multiple texts as the image’s attribute label. For example, if the image has two captions (1) “*the woman is smiling*” and (2) “*the woman has blond hair*,” the attribute label for this image is (“*woman*”, “*smiling*,” and “*blond hair*”). Based on these (image, binary attribute label) pairs, we train the network with binary cross-entropy loss. After training the network, we obtain an image’s pseudo-ground-truth mask based on its attention map (Fig. 4 in [17]). We use Otsu method [38] to threshold the attention map as the final pseudo mask ground-truth. Examples of pseudo-ground-truth mask are shown in Fig. 11.

A.5. Finetune CLIP

We finetune the last few layers of CLIP. Specifically, we finetune the last visual resblock, “ln_post,” “proj,” the last text transformer resblock, “ln_final,” “text_projection,” and “logit_scale” in CLIP. Following [39], we use AdamW [36] optimizer and 5×10^{-4} learning rate.

When finetuning CLIP for the *CLIP-guided Contrastive Loss* (Eq. 1), the objective function for finetuning is contrastive loss defined in [47], where we use the (real image, text) pairs from the training split of the dataset for computing the contrastive loss.

As reported by Zhang *et al.* [71], using the same model in training and testing can skew the R-Precision results. To

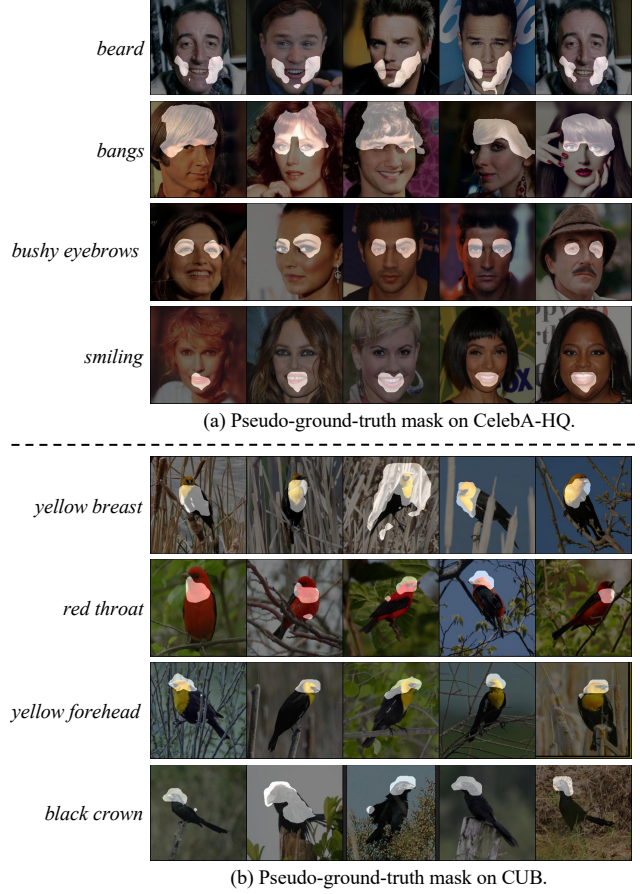


Figure 11. Pseudo-ground-truth masks generated by [17] on CelebA-HQ [18] and CUB [61] datasets. The pseudo-ground-truth mask of the each attribute (e.g., *beard*) is highlighted in white.

alleviate this issue, for computing R-Precision results, we use a CLIP model that is different from the one used in training. We use the contrastive loss to finetune CLIP on the whole dataset (both training and testing splits), which is different from the CLIP used in training (finetuned on the training split only).

When finetuning CLIP for the *Semantic Matching Loss* (Eq. (4)), the objective function for finetuning is binary cross-entropy loss. Concretely, the image’s predicted probability of an attribute is computed by $\text{sigmoid}(\tau \cdot \cos(E_{\text{CLIP}}^{\text{img}}(\mathbf{I}), E_{\text{CLIP}}^{\text{text}}(\mathbf{t}^a)))$. Here, \mathbf{I} denotes an image. \mathbf{t}^a denotes an attribute. τ is the “logit_scale” parameter in CLIP optimized during finetuning. The predicted probability is used in binary cross-entropy to compute the loss.

B. Ablation Studies of Text-to-Image

We show more ablation studies results of text-to-image synthesis.

	R-Precision \uparrow	FID \downarrow
w/o CLIP-guided Contrastive Loss	0.488	17.06
w/o norm penalty	0.736	<u>25.75</u>
w/o Spatial Constraint	0.607	17.45
w/o Compositional Attribute Adjustment	0.594	17.59
w/o finetune CLIP	<u>0.344</u>	17.79
Full Model	0.625	17.46

Table 4. Ablation study of StyleT2I on CelebA-HQ [18] dataset. Top-2 results are bolded and the worst results are underlined.

dataset	threshold (θ)	R-Precision \uparrow	FID \downarrow
CelebA-HQ	8 (min)	0.625	17.46
	16 (mean)	0.815	21.35
	31 (max)	0.801	25.77
CUB	8 (min)	0.264	20.53
	20 (mean)	0.395	22.41
	39 (max)	0.375	26.97

Table 5. Ablation study on the threshold of *norm penalty* (θ in Eq. 2). Here, “min”, “mean”, and “max” stand for the minimum, average, and maximum ℓ_2 norm of two randomly sampled latent codes of the pretrained StyleGAN.

Results on CelebA-HQ We show the ablation study results on CelebA-HQ dataset in Tab. 4. The results are consistent with the ablation study results on CUB dataset in Tab. 3, which further proves the effectiveness of each component of StyleT2I.

Threshold of norm penalty (θ) We conduct an ablation study on different threshold values (θ) of norm penalty (Eq. (2)). To better decide the threshold used for norm penalty, we compute the minimum (min), mean, and maximum (max) ℓ_2 norm between two random latent codes sampled from $\mathcal{W}+$ space of StyleGAN (sampling from $\mathcal{W}+$ space is performed by feeding the sampled Gaussian noise to the “Mapping Network” in StyleGAN). We found that the minimum ℓ_2 norm in StyleGAN trained on CelebA-HQ and CUB datasets are 8.2 and 8.9, respectively. Therefore, we choose $\theta = 8$ in our experiment to force the *Text-to-Direction* and *Attribute-to-Direction* modules find the direction with the smallest norm. As results shown in Tab. 5, although larger θ can increase R-Precision results, it also renders worse image quality (larger FID values). Hence, using $\theta = 8$ strikes a nice balance between image-text balance and image quality.

Alternatives to norm penalty We also tried other alternatives to improve image quality. One way is using the discriminator loss—making the synthesized image fool a discriminator. Another approach is using the perceptual loss to minimize the feature distance between the synthesized and real images. As the results shown in Tab. 6, our *norm*

dataset	method for image quality	FID \downarrow
CelebA-HQ	discriminator	32.83
	perceptual loss	24.98
	<i>norm penalty (Ours)</i>	17.46
CUB	discriminator	26.25
	perceptual loss	29.49
	<i>norm penalty (Ours)</i>	20.53

Table 6. Ablation study of different methods for improving image quality.

Method	R-Precision \uparrow	FID \downarrow
ControlGAN	0.498	17.36
DAE-GAN	0.546	19.24
TediGAN-A	0.026	12.92
TediGAN-B	0.354	14.19
StyleT2I (Ours)	0.635	15.60

Table 7. Results on CelebA-HQ’s standard split.

penalty is the most effective way to ensure the image quality, while other approaches produce much higher FID values (*i.e.*, worse image quality results).

Training Stage Regularization We create an alternative to *Compositional Attribute Adjustment*—“Training Stage Regularization.” While our *Compositional Attribute Adjustment* adjusts the sentence direction during the inference stage, “Training Stage Regularization” maximizes the cosine similarity between the sentence direction and attribute directions, *i.e.*, $\max \sum_i \cos(\mathbf{s}, \mathbf{a}_i)$, which is added as an additional loss to Eq. 3 to regularize the *Text-to-Direction* module during the training stage. The results comparing the “Training Stage Regularization” and *Compositional Attribute Adjustment* are shown in Tab. 8. Two methods achieve similar FID results. However, our *Compositional Attribute Adjustment* achieves better R-Precision results than “Training Stage Regularization.” We believe the reason is that regularizing during the training stage only helps for seen attribute compositions in the training set, which cannot ensure the correct attribute prediction during the inference stage. Therefore, our proposed *Compositional Attribute Adjustment* can better improve the image-text alignment by adjusting the results during the inference stage for text with unseen attribute compositions.

Different \mathbf{z} We sample three different \mathbf{z} for each text to compute the standard deviation of R-Precision, which is 0.008, proving that \mathbf{z} does not have a significant effect on the image-text alignment. The synthesized images of the same text in various \mathbf{z} in Fig. 12, proving the diversity of the synthesis results.

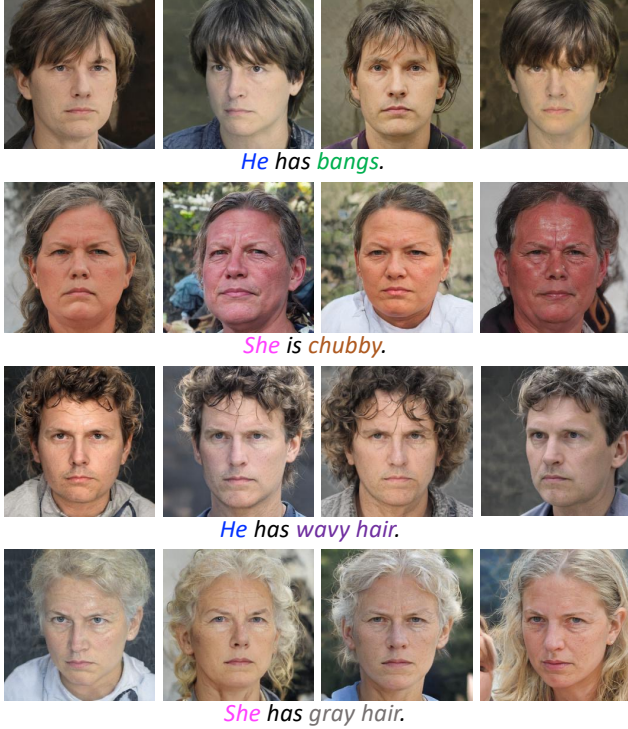


Figure 12. Diverse results when sampling four different \mathbf{z} .

dataset	method	R-Precision \uparrow	FID \downarrow
CelebA-HQ	Training Stage Regularization	0.604	17.56
	<i>Compositional Attribute Adjustment</i>	0.625	17.46
CUB	Training Stage Regularization	0.256	19.48
	<i>Compositional Attribute Adjustment</i>	0.264	20.53

Table 8. Ablation study of *Compositional Attribute Adjustment*. “Training Stage Regularization” stands for using attribute directions to supervise the the sentence direction during the training stage, which can be regarded as an alternative method to *Compositional Attribute Adjustment* that uses attribute directions to adjust sentence direction during the inference stage.

Results on CelebA-HQ’s standard split We also show the results on the CelebA-HQ’s standard testing split, *i.e.*, not the test split that we created for the evaluation of compositionality (Sec. 6.1), in Tab. 7. Most of the results are better than the results on the new split (Tab. 1) because of the overlap between train and test splits that allows the models to cheat.

C. Ablation Studies of Identifying Attribute Directions

We further conduct more ablation studies of identifying attribute directions on CelebA-HQ dataset. To evaluate the identified attribute directions, we train a ResNet-18 classifier with the ground-truth attribute labels (*i.e.*, not the labels extracted from text) as the attribute classifier. We use this

	Attribute Accuracy \uparrow
w/o <i>Spatial Constraint</i>	0.827
w/ <i>Spatial Constraint</i>	0.871

Table 9. Ablation study of *Spatial Constraint* for identifying attribute directions on CelebA-HQ dataset.

margin	Attribute Accuracy
0.1	0.577
0.5	0.761
1	0.871
5	0.881
10	0.875
20	0.873

Table 10. Ablation study on the margin (α) of *Semantic Matching Loss* on CelebA-HQ dataset. The accuracy results are not sensitive to the value of margin when $\alpha \geq 1$.

attribute classifier to evaluate the synthesized positive and negative images generated from *Attribute-to-Direction* module (Fig. 3). For the positive image, its attribute ground-truth is positive. For the negative image, its attribute ground-truth is negative. We compute *Attribute Accuracy* based on the attribute classifier’s prediction and ground-truth. Higher *Attribute Accuracy* indicates a more accurate attribute direction.

Spatial Constraint The results of the ablation study on *Spatial Constraint* are shown in Tab. 9, which proves that *Spatial Constraint* can help the *Attribute-to-Direction* module find more accurate attribute directions by leveraging the intended region from pseudo-ground-truth mask.

Margin of Semantic Matching Loss (α) We conduct the ablation study on the margin (α) of *Semantic Matching Loss* (Eq. (4)). The results in Tab. 10 show that the results are converged when $\alpha \geq 1$. We choose $\alpha = 1$ in the main experiments.

Alternative to Spatial Constraint An alternative approach to improve disentanglement among different attributes is encouraging different attribute directions to be orthogonal with each other in the latent space [53]. Therefore, we create an alternative approach by minimizing $\sum_i \sum_j \frac{\mathbf{a}_i^T \mathbf{a}_j}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}$ when training the *Attribute-to-Direction* module. The results in Tab. 11 show that this alternative approach hurts the accuracy performance compared with only using the *Semantic Matching Loss*. In contrast, our *Spatial Constraint* can greatly improve the accuracy results.

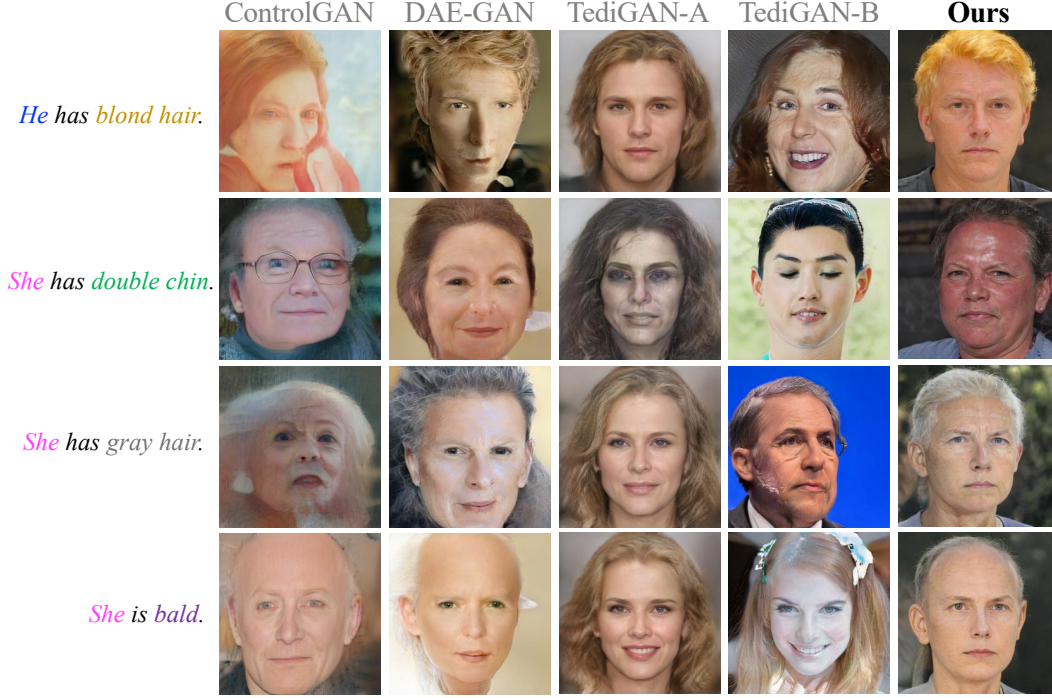


Figure 13. More examples of synthesis results where the input text describes underrepresented compositions of attribute on CelebA-HQ dataset.

	Attribute Accuracy
<i>Semantic Matching Loss</i> only	0.827
w/ $\min \sum_i \sum_j \frac{\mathbf{a}_i}{\ \mathbf{a}_i\ _2}^T \frac{\mathbf{a}_j}{\ \mathbf{a}_j\ _2}$	0.809
w/ <i>Spatial Constraint</i>	0.871

Table 11. Comparison between *Spatial Constraint* and an alternative approach $\min \sum_i \sum_j \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2}^T \frac{\mathbf{a}_j}{\|\mathbf{a}_j\|_2}$ for disentanglement on CelebA-HQ dataset. *Spatial Constraint* achieves better results.

Alternative to Semantic Matching Loss—Contrastive Loss Since the *Text-to-Direction* module and *Attribute-to-Direction* module share some similarity, one may wonder if it is feasible to use the contrastive loss to train the *Attribute-to-Direction*. To this end, we adapt our *CLIP-guided Contrastive Loss* for *Attribute-to-Direction* module by replacing the text input with attribute input, which attracts the embeddings of paired synthesized image and attribute and repels the embeddings of mismatched pairs.

The results of comparing this alternative method and *Semantic Matching Loss* are shown in Tab. 12. The contrastive loss achieves poorer performance for identifying attribute directions. The reason is that we should not repel the embeddings mismatched (image, attribute) pairs. For example, we should not repel the embedding of an “smiling” image against “man” attribute when the random latent code \mathbf{z} can be used to synthesize a male face image. Therefore, our

	Attribute Accuracy
Contrastive Loss + <i>Spatial Constraint</i>	0.669
<i>Semantic Matching Loss</i> + <i>Spatial Constraint</i>	0.871

Table 12. Ablation study of *Semantic Matching Loss* for identifying attribute directions on CelebA-HQ dataset.

Semantic Matching Loss can identify the attribute directions better since it does not repel the embeddings of mismatched (image, attribute) pairs.

Local Direction vs. Global Direction Our *Attribute-to-Direction* module predicts the attribute direction conditioned on both input attribute and random latent code \mathbf{z} . One may wonder if conditioning on the random latent code is necessary. Following the terms defined by Zhuang *et al.* [77], we call the attribute direction conditioned on the random latent code as “local direction,” and we name the attribute direction only conditioned on the attribute (*i.e.*, not conditioned on random latent code) as “global direction.” The results comparing local direction and global direction are shown in Tab. 13. The global direction, which predicts a single direction for an attribute globally, achieves poor attribute accuracy results. In contrast, our local direction method, which takes the random latent code into the consideration, can more accurately predict the attribute direction.

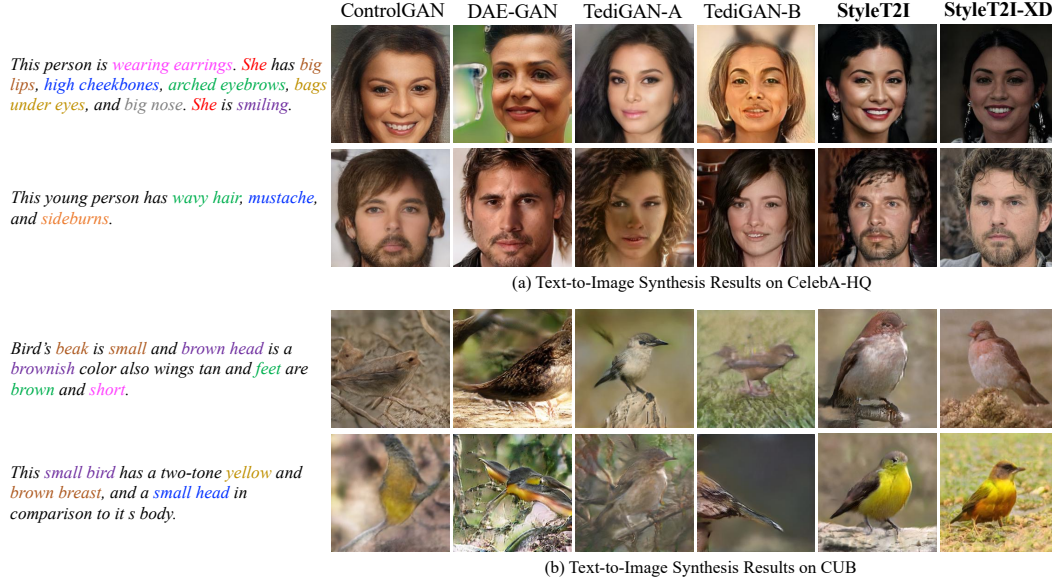


Figure 14. More examples of text-to-image synthesis results.

	Attribute Accuracy
global direction	0.764
local direction (Ours)	0.871

Table 13. Ablation study of global direction vs. local direction for identifying attribute directions on CelebA-HQ dataset.

D. More Qualitative Results

Underrepresented Compositions More examples of synthesis results where the input texts describe underrepresented compositions of attributes are shown in Fig. 13. Our method can more accurately synthesize the image for underrepresented attribute compositions with high image fidelity.

Text-to-Image Results More examples of text-to-image synthesis results are shown in Fig. 14. Our method can synthesize images conditioned on the text describing unseen attribute compositions with better image-text alignment and higher image quality.

Norm Penalty More examples of the ablation study on *norm penalty* are shown in Fig. 15, which proves that *norm penalty* can effectively improve the image quality.

Compositional Attribute Adjustment More examples of the ablation study on *Compositional Attribute Adjustment* (CAA) are shown in Fig. 16, which demonstrates that CAA can automatically identify the wrong attribute predictions and effectively correct them during the inference stage to improve the compositionality.

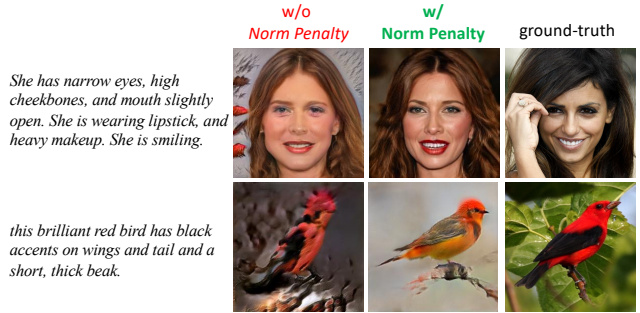


Figure 15. More examples of the ablation study on *norm penalty*.

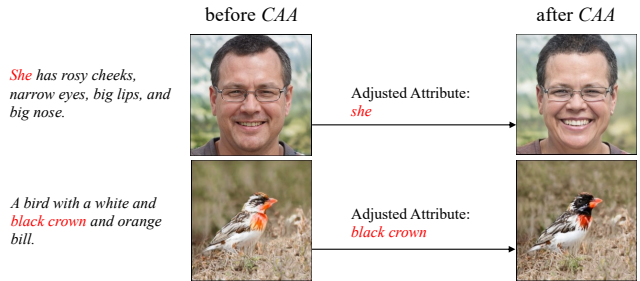


Figure 16. More examples of the ablation study on *Compositional Attribute Adjustment* (CAA).

E. User Study

On each dataset, we randomly sample 20 sentences from the testing split to synthesize the images for the user study. We invite 12 participants to evaluate the image-text alignment and the image quality.

We request the participants to read a guideline before conducting the user study. For evaluating the image-text

1. Please rank the alignment between the image and the given caption (1 to 5 means the "worst" to the "best"). *

She is wearing earrings. She is young and has bags under eyes, and big lips.












Image (a)
Image (b)
Image (c)
Image (d)
Image (e)

	1 (worst)	2	3 (medium)	4	5 (best)
Image (a)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (b)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (c)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (d)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (e)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1. Please rank the image quality (1 to 5 means the "worst" to the "best"). *












Image (a)
Image (b)
Image (c)
Image (d)
Image (e)

	1 (worst)	2	3 (medium)	4	5 (best)
Image (a)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (b)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (c)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (d)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Image (e)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(a) User interface for ranking image-text alignment.
(b) User interface for ranking image quality.

Figure 17. User interface for user study.

alignment on face images, our guideline clarifies that the words like “woman,” “man,” “she,” “he” denote the visually perceived gender, which does not imply one’s real gender identity. Since participants may not be familiar with some terms in the birds image domain, we provide Fig. 2 in [61], an illustration of fine-grained bird part names (e.g., nape), in the guideline of the user study to help participants better understand the text.

We use Google Form to collect the user study results. The user interface for the user study is shown in Fig. 17. The method names are not shown in the user interface. In each question, the order of images generated from different methods is shuffled.

The user study in this paper follows the research protocol, whose master study received the exempt determination from Institutional Review Board (IRB).

F. Discussion

F.1. Limitations and Future Research Directions

We honestly list some limitations of our work and discuss some promising future research directions.

First, our attribute extraction approach (Appendix A.3) is limited by assuming that adjectives and nouns in the text can imply the attribute, which cannot be generalized to texts describing more complex relations in the image. For example, the text “*the earring on the left is bigger than the earring on the right*,” describes a relative relation (e.g., “*bigger*”), which cannot be expressed as an attribute.

Second, based on StyleGAN, StyleT2I focuses on synthesizing fine-grained images in face and bird domains, where StyleGAN has shown a great capability of synthesizing high-fidelity images. However, our initial experiment finds that StyleGAN cannot synthesize high-quality com-

plex scene images from MS-COCO [7,33] dataset, which limits our method to focus on fine-grained single-object image domains, e.g., faces and birds. Future works can study how to leverage pretrained scene image generators (e.g., SPADE [40]) to perform text-to-image synthesis.

Third, in terms of *Spatial Constraint*, the pseudo-ground-truth masks for some images are not accurate, which introduces label noises for *Spatial Constraint*. Future work can leverage some recent semi-supervised methods to obtain the pseudo-ground-truth mask for *Spatial Constraint*. For example, by only annotating a few images, [74] uses StyleGAN to synthesize high-quality images with pseudo-ground-truth masks, which can be used as an alternative to the weakly-supervised method [17] used in this work.

F.2. Potential Negative Societal Impacts

Since StyleT2I can synthesize high-fidelity images, a malicious agent may use our model as a deepfake technology for unintended usage. To mitigate this issue, we ask the users to agree to the ethics terms when releasing the model. Overall, StyleT2I improves the compositionality of text-to-image synthesis, which can better synthesize images for text containing underrepresented attribute compositions, e.g., “*he is wearing lipstick*.” Therefore, we believe that StyleT2I contributes to reducing the negative societal impact compared with previous text-to-image synthesis methods.